

UNITED STATES PATENT APPLICATION

of

BRADLEY H. CLEMENTS

and

JOHN A. FORD

for

**METHODS AND SYSTEMS FOR TRACKING STORAGE RESOURCES
ASSOCIATED WITH A DOCUMENT DISTRIBUTION SYSTEM**

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to the storage of data with a document storage repository. More specifically, the present invention relates to methods and systems by which storage resources are tracked and allocated between one or more users that have an account or storage space within the document storage repository.

2. **Background Technology**

Almost since the inception of the computer, users have interacted and performed useful work on a computer through specialized programs designed to perform various tasks. These specialized programs, often referred to as application programs, allow users to perform desired tasks such as word processing, spreadsheet creation, data entry, and so forth. When a user wishes to perform a certain task, the user starts the execution of an application program and performs all interaction with the computer through the application program. For instance, when a user wishes to create a word processing document, the user initiates execution of the word processing program and creates or edits the word processing document through the word processing program. Once a document is created or edited the document can be saved for later retrieval, such as upon a hard disk, a removable magnetic disk, or the like.

With advances in the Internet and the increased connectivity of computers throughout the world, storage of documents created by different application programs has become an important concern to individuals and businesses a like. For instance, in today's business environment, several individuals may need to collaboratively work on a given

1 project at the same time. This is the case even if the numerous individuals are in different
2 buildings, states, or countries.

3 As the scope and connectivity of the Internet increases, businesses have attempted
4 to harness the increased connectivity and simplicity of communication achievable through
5 the Internet to become “connected” with individuals using their services and products. For
6 instance, many banks provide online banking services, while some businesses provide
7 online shopping or purchasing. The expansive characteristics of the Internet provide the
8 possibility for “paperless” communication between businesses and their customers,
9 resulting in the potential for reduced costs and an increased number of services that can be
10 provided. Unfortunately, to create such a “paperless” world, large data storages are need
11 that can be easily accessed by multiple individuals.

12 In an attempt to provide the necessary connectivity and storage capabilities
13 required by individuals, and more so businesses, various systems have been developed to
14 store data, while providing multi-user access to such data. For instance, one such system is
15 disclosed in United States Patent No. 6,182,080 B1, entitled “System, Method and
16 Computer Program Product for Storage of a Plurality of Documents within a Single File,”
17 the disclosure of which is incorporated herein by reference. This system provides a
18 mechanism whereby a single file, termed a container file, is stored at a central data
19 repository. Multiple documents or files are stored within the container file, no matter the
20 type of application program needed to “open” or access the document. Each container file
21 can include attributes that define those individuals that can access, read, write, or otherwise
22 manipulate the data stored therein. Consequently, an owner of a container file can provide
23 access to the container file, or individual documents within the container file, whether or
24

1 not the individual accessing the document or file is in the same building, state or country.
2 Other storage systems have been developed to allow individuals and businesses to create a
3 central data repository that is accessible by multiple users.

4 Unfortunately, there remain a number of problems with existing document storage
5 repositories. An important aspect of the various data repository type systems is the manner
6 by which data storage resources are allocated and tracked. Typically, data repository
7 providers set an upper limit to the amount of data that can be stored within an individual or
8 business data store or storage space, *e.g.*, ten megabytes. Each user, whether individual or
9 business, pays the data repository provider a defined sum of money per the account owned
10 or the amount of storage used by the individual or business. Consequently, each user can
11 store any number of documents, whether word processing, spreadsheet, graphics, and the
12 like, up to the limit of the data storage or user account. Some other data repository
13 providers track the amount of data storage used at a defined time, *e.g.*, a provider checks
14 the number of documents stored within a user's data storage on a weekly or monthly basis.

15 No matter which technique is used, each document within the user's data store or
16 account is tracked and included in the user's storage total. This is true even if the user
17 does not desire the documents received and stored within the user's data store. For
18 instance, assuming that the user has a ten megabyte data store, in the event that the user
19 receives a one megabyte advertisement from some other user, such as an advertisement
20 distributor, the available space within which the user can store other documents is reduced
21 by one megabyte to nine megabytes. In the event that the user receives nine additional
22 advertisements from the same or other advertisement distributors, each being one
23 megabyte in size, the user's data storage total is ten megabytes. The user, therefore, is

1 unable to store any additional documents until one or more of the advertisements is deleted
2 or the size of the user's data store is increased. Further, if receipt of such advertisements
3 occurs within the usage calculation period, the user will be charged for the use of ten
4 megabytes of space, while the user does not use any of the data space for documents they
5 wish to access or store.

6 It would therefore be an advance to provide a document storage repository that is
7 configured to accommodate the delivery of certain types of documents to different users
8 without the storage of the document affecting the overall storage totals for the recipient
9 user.

SUMMARY AND OBJECTS OF THE INVENTION

It is an object of the present invention to provide methods and systems where the resources used by one user are allocated to another user.

Another object of the present invention is to provide methods and systems where resources utilized by a user to store a document within the user's account are allocated to the author of the stored document instead of the user storing the document.

It is yet another object of the present invention to provide methods and systems for identifying the authorship of a document and/or container file stored within a user's account.

Still another object of the present invention is to provide methods and systems where a document distributor can deliver documents to various users without the need for the document distributor to deliver the document via a delivery container file.

It is another object of the present invention to provide methods and systems where a user can create documents and store the same within a container file and optionally allow multiple users to access the created document.

Yet another object of the present invention is to provide methods and systems where resources are allocated to a document distributor based upon the number of documents that the document distributor delivers to one or more users.

Still another object of the present invention is to provide methods and systems where resources are allocated to a document distributor based upon the amount of storage utilized by those documents stored within user accounts that are authored by the document distributor.

1 As discussed above, many existing document storage repositories are capable of
2 storing amounts of data. These document storage repositories allow a resource for
3 multiple users to store and access documents. Unfortunately, current document storage
4 repositories are incapable of selectively calculating the resources utilized by each user,
5 where the usage by some users is allocated to some other user. It would, however, be
6 desirable if a document storage repository could shift resource usage from one user to
7 another user based upon the data or information stored within a user's account. To
8 overcome the currently existing limitations to allocation of resources between different
9 users, the present invention disclosed herein provides systems and methods for tracking the
10 allocation of resources within a document storage repository, where certain amounts of a
11 user's storage total is shifted or applied to the storage total of another user.

12 According to one aspect, the present invention includes a document storage
13 repository having one or more user accounts. Each user account provides a storage space
14 where a user can store documents, whether as a stand-alone document or within a container
15 file. The user accounts can include a number of attributes and/or properties that define
16 various characteristics of the user account and the owner of such an account. For instance,
17 the user account can include attributes that define the user's name, address, electronic mail
18 address, or the like. Further, an attribute or other identifier can designate a user as a
19 document distributor.

20 These document distributors can distribute a large number of documents to multiple
21 users, however the storage used by such documents is allocated to the document distributor
22 and not to the recipient of the document. Consequently, the present invention provides a
23 mechanism whereby documents created by document distributors and subsequently shared

24

1 with or delivered to other users are not counted towards the storage totals of the recipients.
2 Instead, the document distributors are allocated the resources used by the documents they
3 deliver. This is true whether or not the document distributor delivers or shares a single
4 document or a container file that includes one or more documents.

5 The container file represents a single storage file within which a variety of different
6 documents can be stored. One or more data fields within the document storage repository
7 can represent this storage file. The container file can simultaneously store documents that
8 are accessible by a word processing application, a spreadsheet application, a graphics
9 application, or the like. The term “container file” defines a number of different types of
10 storage files, including a sharing container file that can be used to share one or more
11 documents between multiple users, a delivering container file that can be used to deliver
12 one or more documents from one user to another, or the like.

13 As mentioned above, each user account can include numerous attributes and
14 properties. Similarly, each document and container file can include attributes and
15 properties that identify certain characteristics of the document and container file. Each of
16 the attributes and/or properties for the user accounts, documents, and container files can be
17 included within one or more data fields. Accordingly, each document and container file
18 can include, but is not limited to, a name attribute, a date created attribute, a date modified
19 attribute, an access list, an author attribute, and optionally a “free storage” attribute,
20 optionally contained within one or more data fields. Whenever the document or the
21 container file is created by a document distributor, *i.e.*, a user with a specified value or
22 pointer to some other attribute, the free storage attribute is added to the document or
23 container file attributes indicating that it was created by the document distributor. This

24

1 free storage attribute is affixed whether the document is created manually through a user
2 interface or via an automated process using an appropriate application program interface
3 (API) associated with the document storage repository. This free storage attribute remains
4 with the document or container file even if the document or container file is transferred to
5 another user account or shared with another account.

6 Alternatively, an author attribute of the document and container file can point to the
7 user attributes of the author of the document and container file, and more specifically, to a
8 document distributor attribute that identifies the user as a document distributor or a general
9 user.

10 According to another aspect of the present invention, each time a document or
11 container file is created by a document distributor, a transaction is written to a log stored at
12 the document storage repository. This log tracks substantially all actions taken by each
13 user and the number and/or size of the documents and/or container files delivered or shared
14 by the document distributor and other users. In one configuration of the present invention,
15 such data stored within the log can be stored as one or more data fields and imported into a
16 billing system to compute charges for the document distributor and other users. Further,
17 the document storage repository can use the log to compute a user's storage total, while
18 optionally analyzing the document distributor attributes for each user, document, and/or
19 container file, and/or each free storage attribute to exclude certain documents and/or
20 container files from one user and include the same documents and/or container files in the
21 total for an associated document distributor. The resultant determination of where
22 documents and/or container files are to be associated can optionally be stored in one or
23
24

1 more data fields accessible by various modules or components of the document storage
2 repository.

3 These and other features of the present invention will become more fully apparent
4 from the following description and appended claims, or can be learned by the practice of
5 the invention as set forth hereinafter.

BRIEF DESCRIPTION OF THE DRAWINGS

2 In order to describe the manner in which the above-recited and other advantages
3 and features of the invention can be obtained, a more particular description of the invention
4 briefly described above will be rendered by reference to specific embodiments thereof that
5 are illustrated in the appended drawings. Understanding that these drawings depict only
6 typical embodiments of the invention and are not therefore to be considered limiting of its
7 scope, the invention will be described and explained with additional specificity and detail
8 through the use of the accompanying drawings in which:

9 Figure 1 illustrates an exemplary system that provides a suitable operating
10 environment for the present invention;

11 Figure 2 is a top level diagram illustrating another exemplary system that provides
12 a suitable operating environment for the present invention;

13 Figure 3 is more detailed representation of the top level diagram of Figure 2
14 illustrating one embodiment of the present invention;

15 Figure 4 is a diagram illustrating a structured storage of one embodiment of the
16 present invention;

17 Figure 5 is a flow diagram representing a method for uploading a document into the
18 structured storage of one embodiment of the present invention;

19 Figure 6 is a schematic representation of the transfer and retrieval of data from
20 various modules of the structured storage of one embodiment of the present invention;

21 Figure 7 is a flow diagram representing a method for calculating resource allocation
22 of data within the structured storage of one embodiment of the present invention;

1 Figure 8 is a flow diagram representing another method for calculating resource
2 allocation of data within the structured storage of one embodiment of the present
3 invention; and

4 Figure 9 is a flow diagram representing a method for calculating resource allocation
5 of data within the structured storage of one embodiment of the present invention and
6 creating a billing statement based upon the calculated resource allocation.

DETAILED DESCRIPTION OF THE INVENTION

The present invention extends to both methods and systems for tracking the allocation of resources within a document storage repository. Through tracking storage resource usage, the present invention can identify the amount of storage space utilized by each user, *i.e.*, identify the size and number of documents and/or files accessible to a particular user. In addition, the present invention facilitates analysis of each file or document to identify the author or creator of the file or document. Consequently, the present invention tracks whether a specialized user, termed a “document distributor” authored the stored files or documents, and based upon the tracked information, the present invention can accurately calculate the allocation of storage resources for each user and optionally calculate costs associated with the storage resources used by each user.

The embodiments of the present invention can comprise a special purpose or general-purpose computer that includes computer hardware, as discussed in greater detail below. Embodiments within the scope of the present invention also include computer-readable media for carrying or having computer-executable instructions or data structures, including one or more data fields, stored thereon. Such computer-readable media can be any available media that can be accessed by a general purpose or special purpose computer. By way of example, and not limitation, such computer-readable media can comprise RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other medium that can be used to carry or store desired program code means in the form of computer-executable instructions or data structures and that can be accessed by a general purpose or special purpose computer. When information is transferred or provided over a network or another communications

1 connection (either hardwired, wireless, or a combination of hardwired or wireless) to a
2 computer, the computer properly views the connection as a computer-readable medium.
3 Thus, any such connection is properly termed a computer-readable medium.
4 Combinations of the above should also be included within the scope of computer-readable
5 media. Computer-executable instructions comprise, for example, instructions and data,
6 optionally stored in data fields, which cause a general purpose computer, special purpose
7 computer, or special purpose processing device to perform a certain function or group of
8 functions.

9 Figure 1 and the following discussion are intended to provide a brief, general
10 description of a suitable computing environment in which the invention can be
11 implemented. Although not required, the invention will be described in the general context
12 of computer-executable instructions, such as program modules, being executed by
13 computers in network environments. Generally, program modules include routines,
14 programs, objects, components, data structures, including data fields, etc. that perform
15 particular tasks or implement particular abstract data types. Computer-executable
16 instructions, associated data structures, and program modules represent examples of the
17 program code means for executing steps of the methods disclosed herein. The particular
18 sequence of such executable instructions or associated data structures, optionally including
19 one or more data fields, represents examples of corresponding acts for implementing the
20 functions described in such steps.

21 Those skilled in the art will appreciate that the invention can be practiced in
22 network computing environments with many types of computer system configurations,
23 including personal computers, hand-held devices, multi-processor systems,
24

1 microprocessor-based or programmable consumer electronics, network PCs,
2 minicomputers, mainframe computers, and the like. The invention can also be practiced in
3 distributed computing environments where tasks can be performed by local and remote
4 processing devices that are linked (either by hardwired links, wireless links, or by a
5 combination of hardwired or wireless links) through a communications network. In a
6 distributed computing environment, program modules can be located in both local and
7 remote memory storage devices.

8 With reference to Figure 1, an exemplary system for implementing the invention
9 includes a general-purpose computing device in the form of a conventional computer 20,
10 including a processing unit 21, a system memory 22, and a system bus 23 that couples
11 various system components including the system memory 22 to the processing unit 21.
12 The system bus 23 can be any of several types of bus structures including a memory bus or
13 memory controller, a peripheral bus, and a local bus using any of a variety of bus
14 architectures. The system memory includes read only memory (ROM) 24 and random
15 access memory (RAM) 25. A basic input/output system (BIOS) 26, containing the basic
16 routines that help transfer information between elements within the computer 20, such as
17 during start-up, can be stored in ROM 24.

18 The computer 20 can also include a magnetic hard disk drive 27 for reading from
19 and writing to a magnetic hard disk 39, a magnetic disk drive 28 for reading from or
20 writing to a removable magnetic disk 29, and an optical disk drive 30 for reading from or
21 writing to removable optical disk 31 such as a CD-ROM or other optical media. The
22 magnetic hard disk drive 27, magnetic disk drive 28, and optical disk drive 30 are
23 connected to the system bus 23 by a hard disk drive interface 32, a magnetic disk drive-

24

1 interface 33, and an optical drive interface 34, respectively. The drives and their
2 associated computer-readable media provide nonvolatile storage of computer-executable
3 instructions, data structures, program modules and other data for the computer 20.
4 Although the exemplary environment described herein employs a magnetic hard disk 39, a
5 removable magnetic disk 29 and a removable optical disk 31, other types of computer
6 readable media for storing data can be used, including magnetic cassettes, flash memory
7 cards, digital versatile disks (DVDs), Bernoulli cartridges, RAMs, ROMs, and the like.

8 Program code means comprising one or more program modules can be stored on
9 the hard disk 39, magnetic disk 29, optical disk 31, ROM 24 or RAM 25, including an
10 operating system 35, one or more application programs 36, other program modules 37, and
11 program data 38. A user can enter commands and information into the computer 20
12 through keyboard 40, pointing device 42, or other input devices (not shown), such as a
13 microphone, joy stick, game pad, satellite dish, scanner, or the like. These and other input
14 devices are often connected to the processing unit 21 through a serial port interface 46
15 coupled to system bus 23. Alternatively, the input devices can be connected by other
16 interfaces, such as a parallel port, a game port or a universal serial bus (USB). A monitor
17 47 or another display device is also connected to system bus 23 via an interface, such as
18 video adapter 48. In addition to the monitor, personal computers typically include other
19 peripheral output devices (not shown), such as speakers and printers.

20 The computer 20 can operate in a networked environment using logical connections
21 to one or more remote computers, such as remote computers 49a and 49b. Remote
22 computers 49a and 49b can each be another personal computer, a server, a router, a
23 network PC, a peer device or other common network node, and typically include many or
24

1 all of the elements described above relative to the computer 20, although only memory
2 storage devices 50a and 50b and their associated application programs 36a and 36b have
3 been illustrated in Figure 1. The logical connections depicted in Figure 1 include a local
4 area network (LAN) 51 and a wide area network (WAN) 52 that are presented here by way
5 of example and not limitation. Such networking environments are commonplace in office-
6 wide or enterprise-wide computer networks, intranets and the Internet.

7 When used in a LAN networking environment, the computer 20 is connected to the
8 local network 51 through a network interface or adapter 53. When used in a WAN
9 networking environment, the computer 20 can include a modem 54, a wireless link, or
10 other means for establishing communications over the wide area network 52, such as the
11 Internet. The modem 54, which can be internal or external, is connected to the system bus
12 23 via the serial port interface 46. In a networked environment, program modules depicted
13 relative to the computer 20, or portions thereof, can be stored in the remote memory
14 storage device. It will be appreciated that the network connections shown are exemplary
15 and other means of establishing communications over wide area network 52 can be used.

16 Referring now to Figure 2, depicted is a schematic representation of one illustrative
17 network system 200 within which the methods and systems of the present invention can be
18 incorporated. The system 200 facilitates the storage of documents and other information in
19 files that can be easily exchanged and used by individuals or groups of individuals. For
20 example, a single file can contain multiple different documents that use different
21 application programs to open or access the contents of the documents. These documents
22 and other information can be stored in a single file that can be referred to as a container
23 file. One illustrative configuration of such a container file and a system within which such
24

1 a container file can be used is described in United States Patent No. 6,182,080 B1 to
2 Clements (hereinafter the “Clements patent”), which is incorporated herein by reference.

3 Although it is preferred that each container file and system 200 have a
4 configuration similar to that described in the Clements patent, it will be appreciated by one
5 skilled in the art that various other systems are applicable in performing the functions
6 described herein in light of the teachings contained herein.

7 The system 200 can include one or more document distributors 202a-202n that
8 communicate via network 204 with a secure document storage repository 206. Further,
9 numerous free users 208a-208n and fee-based users 210a-210n can communicate with or
10 access document storage repository 206 via network 204. In this manner, various users
11 can access document storage repository 206 from numerous different remote or local
12 locations via network 204, such as a WAN, a LAN, a wireless network, a packetized
13 network, and the like.

14 Generally, document distributors 202a-202n are a specialized type of user that
15 wishes to distribute documents to one or more of the users having an account with
16 document storage repository 206, such as free users 208a-208n and fee-based users 210a-
17 210n, while limiting or substantially eliminating any increase in the storage totals of those
18 users that receive the document distributors’ correspondence. Stated another way, each
19 document distributor wishes to send correspondence to free users 208a-208n and fee-based
20 users 210a-210n with the knowledge that the size of the correspondence, *i.e.*, the number
21 of bytes forming the correspondence, will not be subtracted from the user’s account or will
22 not be counted against the user when the user’s data storage total is calculated. Typically,
23 document distributors can include brokers, utility companies, and the like that wish to send

24

1 mass mailings, statements, advertisements, and other information to one or multiple
2 individuals having an account with document storage repository 206.

3 In contrast to each document distributor 202a-202n, each free user 208a-208n is
4 considered a general user of document storage repository 206. Consequently, each free
5 user 208a-208n is provided with a defined storage space, *e.g.*, from about one megabyte to
6 about ten megabytes, within which each free user 208a-208n can store various documents
7 whether as stand alone documents or as part of or included with a stored container file. As
8 the name suggests, each free user 208a-208n is provided with a designated storage space at
9 no cost. In the event that a free user 208a-208n exceeds the allotted amount of storage
10 space, the free user 208a-208n can switch to being a fee-based user 210a-210n where the
11 user has to pay for the amount of storage used in excess of the allotted storage space
12 associated with being a free user 208a-208n. Alternatively, a free user 208a-208n may be
13 charged the fee associated with a fee-based user 210a-210n without prorating for the
14 amount of storage in excess of the designated free user allotted storage space.

15 As may be understood from the above, each fee-based user 210a-210n pays a
16 particular fee for use of a designated amount of storage space that is typically greater than
17 the storage space provided to a free user 208a-208n. Optionally, each fee-based user 210a-
18 210n can have access to an unlimited amount of space, so long as the provider of document
19 storage repository 206 believes that a fee-based user 210a-210n is capable of paying the
20 appropriate fee.

21 As used herein and understood by one skilled in the art, the term “user” includes
22 document distributors 202a-202n, free users 208a-208n, and fee-based users 210a-210n.
23 More generally, the term “user” can refer to and include any individual or entity that can

1 access or communicate with document storage repository 206, have an account with
2 document storage repository 206, or otherwise be able to store, retrieve, manipulate, or
3 exploit documents and/or container files stored at document storage repository 206. For
4 example, each user can take the form of an individual operating a workstation, such as
5 computer 20 described above, which can connect with network 204 and enable access to
6 document storage repository 206 and the container files and documents stored therein.
7 Alternatively, each user can take the form of an individual server and/or client or multiple
8 servers and/or clients that access document storage repository 206. In such a case, the
9 servers and/or clients can use an application program interface (API) to directly
10 communicate with document storage repository 206 and the stored container files and
11 documents.

12 Generally, document storage repository 206 includes data storage modules upon
13 which data created by the users can be stored as stand alone documents or as part of one or
14 more container files. Further, document storage repository 206 provides a structured
15 environment for storing and retrieving documents. The document storage repository 206
16 can include a plurality of servers that communicate one with another and store data and
17 other information on hard disks, magnetically readable media, optically read media, and
18 the like. Such servers can be local one to another or remotely located but accessible one to
19 another.

20 In addition to the modules described above, document storage repository 206 can
21 include software applications to provide a graphical user interface (GUI) that facilitates
22 access to the data stored in document storage repository 206. This GUI, such as a web site,
23 can let users create a user account specific and unique to the particular user. Consequently,

1 the GUI can assist an individual to become a free user 208a-208n, a fee-based user 210a-
2 210n or a document distributor 202a-202n with the associated rights and properties. As
3 mentioned above, a free user may be limited in the amount of data that they can store in
4 document storage repository 206, while a fee-based user may be capable of storing a larger
5 amount of data or an unlimited amount of data if they are capable of paying for their
6 storage requirements. Similarly, a document distributor may have an unlimited quantity of
7 storage space within document storage repository 206 and may optionally receive a
8 discounted rate for the storage used or that the document distributor is expected to use. In
9 this manner, document storage repository 206 can store user data, while providing an
10 appropriate GUI to facilitate access to the data stored therein.

11 Further, the software associated with document storage repository 206 can facilitate
12 sharing and delivering of documents and container files between the various users having
13 an account with document storage repository 206. For example, one document distributor
14 202a-202n can store a single document or a group of documents in a container file
15 accessible by a single user. Alternatively, document distributor 202a-202n can store single
16 or multiple documents in a container file accessible by multiple users, such as a shared
17 container file. Optionally, the documents can be delivered from one user to another user
18 through the use of a delivery container file. Although reference is made to the actions of
19 document distributor 202a-202n, it will be understood that any user can share, deliver, and
20 store documents in container files or as stand alone documents. Further, each of the above
21 container files can be considered as separate files and therefore, the discussion herein will
22 use the phrase “container file” as representing a storage container file, a delivery container
23 file, and/or a shared container file.

1 Turning now to Figure 3, a more detailed schematic representation of system 200 is
2 depicted. Specifically, various users, *i.e.*, document distributors 202a-202n, free users
3 208a-208n, and fee-based users 210a-210n are represented collectively instead of
4 individually, and designated by reference numeral 212. Each user 212 can access or “log
5 in” to document storage repository 206 using a workstation 220a-220n. The user 212 can
6 then upload or download one or more documents into container files, whether such
7 container files are accessible only by the user, shared among multiple users, or deliverable
8 to one or more users. These workstations 220a-220n can include computer 20, a personal
9 digital assistant (PDA), a palm computer, a mobile or wireless digital phone, multiple
10 computers, combinations thereof, or the like, so long as the device used facilitates access to
11 document storage repository 206.

12 Each workstation 220a-220n can include various application programs to facilitate
13 user 212 accessing, viewing, and retrieving documents or container files stored at
14 document storage repository 206. For example, each workstation 220a-220n can include a
15 web browser to provide an interface through which user 212 can access or “log in” to
16 document storage repository 206. Numerous web browsers are known to one skilled in the
17 art, including but not limited to, Microsoft® Internet Explorer, Netscape® Navigator or
18 Communicator, or the like. Further, each workstation 220a-220n can include application
19 programs specific to the particular type of document stored at document storage repository
20 206. For instance, such application programs can include, but are not limited to, word
21 processing applications, spreadsheet applications, and graphics applications.

22 In addition to using workstations 220a-220n, user 212 can employ one or more
23 servers 222a-222n that optionally automatically create and deliver documents to document

1 storage repository 206 or to container files within repository 206, and optionally retrieve
2 documents from document storage repository 206 or from container files within repository
3 206. Consequently, servers 222a-222n can communicate with document storage repository
4 206 with or without the interaction of a human user to initiate such communication. This
5 can be achieved, in one illustrative configuration, through use of an API that servers 222a-
6 222n use to directly communicate with the hardware and software modules and
7 components of document storage repository 206.

8 Access to document storage repository 206 is controlled by a control module 230.
9 The control module 230, as implied by its name, manages the flow of data to and from
10 document storage repository 206. Consequently, control module 230 can receive data
11 from and deliver data to user 212. In one configuration, control module 230 includes a
12 control application 232 to perform the above functions. The control application 232 can
13 take the form of a web site with one or more web pages that provide a GUI that user 212
14 can employ to access document storage repository 206. Consequently, control module
15 230, in one embodiment, can include one or more web servers that host the above-
16 described control application 232.

17 Generally, control application 232 can have various configurations so long as it is
18 capable of controlling a user's access to the documents and container files stored within
19 document storage repository 206. The control application 232, therefore, implements
20 various user interfaces and functionality to achieve the various goals of the presently
21 described invention. For instance, control application 232 can be responsible for setting up
22 the structured storage used by each of the container files. One illustrative control
23 application is the NetDocuments™ application provided by NetVoyage Corporation.

1 Other functionality of control application 232 can include aiding users in creating a
2 user account that defines the access level and storage limits for each user 212.
3 Consequently, the user accounts can define particular attributes and properties for each
4 user 212. For instance, when a user account is created, user 212 can provide information
5 about the user's name, address, telephone number, electronic mail message (e-mail),
6 business trademark or logo that is presented to a viewer when a document or container file
7 authored by the user is viewed by a user, document distributor designation, user ID,
8 password, and the like. This information is stored within one or more user files 236a-236n
9 in document storage repository 206 and defines each user as a unique individual or entity
10 with specific properties, attributes, and characteristics. More specifically, the user account
11 information, contained in user files 236a-236n can be stored in a directory module 234
12 communicating with control module 230.

13 In addition to the above, control application 232 can: (i) create and maintain one or
14 more tracking logs; (ii) update some or all of the properties of individual documents and
15 container files; (iii) place documents into and retrieve documents from a container file; (iv)
16 perform security functions; (v) synchronize container files; (vi) transfer container files
17 and/or documents from one user account to another user account, including transfer of
18 documents from one container file to another container file, and so forth.

19 As mentioned above, control application 232 can insert documents into one or more
20 container files and retrieve documents from these container files. In one embodiment,
21 control application 232 can store a document retrieved from document distributor 202a-
22 202n, free user 208a-208n, or fee-based user 210a-210n (Figure 2) in a container file by
23 obtaining the document and placing the document into the appropriate storage structure of

1 the container file such as within a storage module 238. In one configuration, a user can
2 access the web site hosted by control module 230 and through control application 232 and
3 its GUI drag an icon representing a document to a location on the user interface
4 representing a container file or its contents. Such an action can be a signal to control
5 application 232 to obtain the identified document and place it into the identified container
6 file.

7 Alternatively, a user can use an API to automatically insert a document into a
8 container file or insert multiple copies of the same document into multiple container files,
9 whether or not one or more of such container files are accessible by one user or multiple
10 users. For instance, when document distributor 202a-202n is a utility company, service
11 provider, or other business or individual wishing to deliver mass mailings, billing
12 statements, and the like, document distributor 202a-202n can use the API to insert
13 documents into container files of those users 212 desiring to receive such communication
14 and information.

15 In a similar fashion to that described above, users can identify documents in a
16 container file that should be removed from the container file. One skilled in the art will
17 appreciate that other mechanisms can allow a user to identify a particular document that
18 should be placed into or removed from a container file. Similarly, there are appropriate
19 mechanisms for inserting and removing container files from a user's account or storage
20 space designated within document storage repository 206.

21 The control module 230 and consequently control application 232 can
22 communicate with directory module 234 to obtain user information and identify particular
23 container files and documents associated with each user 212. Generally, directory module
24

234 can include lists of data associated with each user, such as a list of user objects or user files 236a-236n, which can be created as the user creates a user account. Also, directory module 234 can include a list of the specialized users termed “document distributors.” Although document storage repository 206 can store separate lists of general users and document distributors, it can be appreciated by one skilled in the art that a single list can be maintained with both general users and document distributors. Consequently, and reiterating, the term “user” can include both general users, *i.e.*, fee users and free users, as well as document distributors.

The directory module 234 can take the form of a database of user information, such as but not limited to user names, addresses, telephone numbers, email addresses, and the like. As part of the present invention, directory module 234 can include a document distributor attribute for each user. The document distributor attribute identifies whether the user is or is not a document distributor and consequently identifies the availability of the above-recited rights and/or obligations. Further, once a user is designated as a document distributor, such as by the document distributor attribute having some value, each document or container file created by such a user is designated as being authored by a document distributor through inclusion of a free storage attribute that remains with the document or container file even if the document or container file is delivered or transferred to another user account or shared with one or more other users. Consequently, as those documents and/or container files are analyzed during resource allocation calculations, document storage repository 206 includes the number and/or size of the documents and/or container files that have such a free storage attribute in the document distributor's data storage total, while document storage repository 206 excludes such documents and

1 container files from the recipient's data storage total, as will be discussed in greater detail
2 hereinafter. In light of the teaching contained herein, one skilled in the art can identify
3 various other manners to identify whether or not a user is a document distributor.

4 As discussed thus far, document storage repository 206 facilitates the storage of
5 data that is uploaded from user 212, *i.e.*, from workstations 220a-220n or servers 222a-
6 222n. Optionally, document storage repository 206 can facilitate the delivery of
7 documents from one user to another, *i.e.*, from one container file to another container file.
8 For instance, when multiple users work on a project that requires all users to have access to
9 the same document, one user can create a container file containing the needed documents
10 and allow each user working on the project to access the documents contained within the
11 container file.

12 Generally, a storage module 238 stores documents and container files in a
13 structured manner to allow efficient retrieval and updating of such documents and
14 container files. The storage module 238 can have various configurations so long as it is
15 capable of storing data in a manner that allows access and retrieval of the data as needed
16 by control module 230. In the illustrated configuration, storage module 238 includes one
17 or more container files 240a-240n and a worklist module 242.

18 As described above, each container file 240a-240n can store a wide variety of
19 information. For example, each container file 240a-240n can be used to store one or more
20 documents 244a-244n that are adapted for use by a specific application program. For
21 instance, one of documents 244a-244n may be opened by a word processing application,
22 while another one of documents 244a-244n may be opened by a spreadsheet application or

23
24

1 graphics application. Various other types of document 244a-244n are known by those
2 skilled in the art in light of the teaching contained herein.

3 In addition to documents 244a-244n, other information can also be stored in each
4 container file 240a-240n. For instance, each container file 240a-240n can store a tracking
5 log, such as a tracking log 246a-246n. Each tracking log 246a-246n can store information
6 that allows actions performed on each container file 240a-240n and/or documents 244a-
7 244n to be recorded. The tracking log 246a-246n provides a mechanism whereby changes
8 and other actions performed by individuals or entities on a container file 240a-240n can be
9 tracked and identified. This can provide important security features to container files
10 240a-240n, documents 244a-244n, and other information contained therein. By recording
11 what actions were taken by which individuals, authorized actions taken by individuals can
12 be identified.

13 Container files 240a-240n can also include various other types of information or
14 data. This information is illustrated in Figure 3 by properties 248a-248n. As explained in
15 greater detail below, properties 248-248n represent information about documents 244a-
16 244n or container file 240a-240n that is useful or desired. Such information can include,
17 for example, the author of a particular document, summary information regarding a
18 particular document, version information for the container file or a document, a free
19 storage attribute for the container file or document, and so forth.

20 According to another aspect of the present invention, storage module 238 can
21 include a worklist module 242. The worklist module 242 is configured to track those
22 container files and documents stored in document storage repository 206 accessible by a
23 user, *i.e.*, which documents or container files are stored by the user either alone or in a user
24

1 authored container file, in a shared container file accessible by the user, or the like.
2 Consequently, worklist module 242 can include a list of users and those documents and
3 container files accessible by those users. Optionally, worklist module 242 can include a
4 list of all container files and documents with pointers to the user currently storing or
5 otherwise having access to the same. The worklist module 242, and the information stored
6 therein, is used when system 200 calculates the resource allocation of stored documents
7 and container files within document storage repository 206, as will be described in detail
8 hereinafter.

9 Generally, storage module 238 can take the form of any and all types of optically-
10 read media, magnetically-read media, and the like. Further, storage module 238 can
11 include flash memory, programmable EPROM, RAM, ROM, removable media, and the
12 like. Although storage module 238 is shown as part of document storage repository 206,
13 one skilled in the art will understand that storage module 238 need not be part of document
14 storage repository 206, but can be remote from document storage repository 206 while
15 being accessible by document storage repository 206.

16 In addition to control module 230 communicating with storage module 238 and
17 directory module 234, control module 230 can communicate with resource allocation
18 module 250. Resource allocation module 250 is adapted to calculate the amount of storage
19 utilized by each user having an account with document storage repository 206. Further,
20 resource allocation module 250 can optionally calculate a cost or bill for each user based
21 upon their usage of the storage resources of document storage repository 206. Various
22 types of modules, including conventional hardware and/or software components, are
23 capable of calculating the storage used by each user.

24

1 As illustrated, intermediate between control module 230 and resource allocation
2 module 250 is temporary file 252. Although temporary file 252 is illustrated as being
3 separate from control module 230, one skilled in the art can appreciate that temporary file
4 252 can be incorporated within control module 230, resource allocation module 250, or in
5 some other file or module of the present invention.

6 The temporary file 252 is adapted to temporarily store information related to the
7 usage of document storage repository 206 by the various users. Temporary file 252 stores
8 information about transactions performed by each user, *i.e.*, how many documents were
9 created by a user, to whom the documents were sent, the amount of storage being utilized
10 by a user, and the like. This information can be collected for a defined period of time, such
11 as a twenty-four hour period. Upon gathering the information for such a period of time,
12 control module 230 delivers the information to resource allocation module 250.
13 Alternatively, resource allocation module 250 can recognize the end of the time period and
14 “pull” or import the information from temporary file 252. Consequently, the information
15 stored in temporary file 252 can be pushed to or pulled from resource allocation module
16 250.

17 Various directory structures for document storage repository 206 and associated
18 container files and documents are applicable for use with the present invention. For
19 instance, as described in United States Patent No. 6,182,080, the directory structure can be
20 implemented using a wide variety of structure store technology, such as but not limited to,
21 Microsoft Corporation’s standard referred to as the OLE compound file standard, IBM
22 Corporation and Apple Computer’s standard referred to as OpenDoc, a JAVA standard
23 called JAVA Beans, Novell Network NDS technology directory services, or other systems

24

1 and standards known to one skilled in the art in light of the teaching contained herein. In
2 addition, structured storage can be implemented using no particular standard at all, simply
3 utilizing standard programming techniques to construct an appropriate structure storage
4 mechanism. However, since utilizing an appropriate standard will make the result and the
5 file more interoperable with other applications and programs, it may be desirable to utilize
6 an appropriate standard.

7 Referring now to Figure 4, an illustrative representation of one storage structure of
8 one embodiment of each container file 240a-240n is presented. The illustrative
9 representation depicts only one container file 240a, however, it is understood that a similar
10 discussion can be made for each container file 240a-240n. As illustrated, container file
11 240a has a root storage 260 that defines the storage structure of container file 240a. The
12 root storage 260 can include streams, files, objects, or attributes that define various
13 characteristics of container file 240a and the information stored in container file 240a.
14 Exactly what information is stored will depend on the particular implementation.

15 As shown, container file 240a can include container properties 262, access
16 properties 264 and a tracking log 266. Throughout this discussion, container
17 properties 262, access properties 264, and tracking log 266 each will be identified as being
18 stored as a single stream, file, object, or the like. However, such reference is made for
19 notational convenience only. Container properties 262, as well as access properties 264
20 and tracking log 266, can be stored in more than one stream, file, object, or the like,
21 depending upon the particular implementation of container file 240a. Optionally, in some
22 configurations, container properties 262, access properties 264, and tracking log 266 can

1 include numerous sub-files, sub-streams, sub-objects, sub-attributes, sub-properties, or the
2 like.

3 The container properties 262 define the various attributes associated with container
4 file 240a. Each of these attributes can have a fixed value, a true or false value, or a pointer
5 to some other list stored in directory module 234, storage module 238, or the like.
6 Numerous attributes are applicable for use as part of container properties 262. For
7 instance, container properties 262 can include a list of the container's title or name, the
8 date when container file 240a was created, a document distributor attribute designating
9 whether the container file 240a was created by document distributor 202a-202n (Figure 2),
10 a free storage attribute identifying whether the container file and/or the documents therein
11 should be included or excluded from the user's calculated storage total, the subject of
12 container file 240a, the author or creator of container file 240a, key words describing
13 container file 240a, a template field, a field indicating who was the last individual to save
14 something in container file 240a, a revision number, total editing time, a last printed field
15 describing when a document within container file 240a was last printed, a creation
16 date/time field, a last saved date/time field, number of pages, number of words, number of
17 characters, a thumbnail or summary field, the name of the application creating container
18 file 240a, a security field, or the like. Note, that not all of these fields can be applicable to
19 all implementations of container file 240a or the other container files of the present
20 invention.

21 In addition to the above, container properties 262 can include information relating
22 to the status of container file 240a, the location of the master copy of container file 240a,
23 the modification sequence of container file 240a, a list of items which are synchronized or
24

1 replicated between various copies of container file 240a, an indication of when container
2 file 240a expires, routing information, mappings of various log information to user names,
3 and a return address information displayed to a user in a graphical view of container file
4 240a.

5 Further, in some configurations, container properties 262 can contain user-defined
6 properties. Consequently, a user can define properties that he or she wishes to track and
7 maintain. These properties can then be stored as container properties 262 and retrieved for
8 various purposes. In one embodiment, the user is allowed to dynamically create, modify,
9 and delete properties and information stored in the properties.

10 The access properties 264 define the various users 212 (Figure 3) that can access
11 the documents and information stored within container file 240a. Specifically, access
12 properties 264 can provide a list of those users 212, whether document distributors 202a-
13 202n (Figure 2), free users 208a-208n (Figure 2), or fee-based users 210a-210n (Figure 2)
14 that can read, write, or otherwise manipulate the documents or data stored within container
15 file 240a. Similarly, tracking log 266 stores a record of the individuals or entities that
16 perform various actions on either container file 240a, the documents stored therein, stand
17 alone documents not associated with container file 240a, combinations thereof, or the like.

18 Although the discussion herein has described separate properties forming part of
19 container properties 262, access properties 264, and tracking log 266, one skilled in the art
20 can appreciate that such properties can be stored in various other locations or distributed
21 among the above-recited properties, logs or other similar files, streams, objects, or the like.
22 For example, the properties described above with reference to container properties 262 can

1 be included in access properties 264 and vice versa. Similarly, portions of those container
2 properties 262 can be included in tracking log 266 and vice versa.

3 In a hierarchical relationship to root storage 260 is an individual storage for each of
4 the documents that will be stored in container file 240a. In Figure 4, three document
5 storage locations are defined and these are labeled document 1 storage 270a, document 2
6 storage 270b, and document N storage 270n. As illustrated in Figure 4, any number of
7 document storages can be created and placed in container file 240a. Document storages
8 270a-270n each include document properties 276a-276n, document contents 278a-278n,
9 and an optional tracking log 280a-280n. Each of document properties 276a-276n contains
10 information relating to the individual documents stored in the respective storage 270a-
11 270n. Although any useful or desired information can be stored in document properties
12 276a-276n, in one embodiment document properties 276a-276n include the document
13 name. The document's name is the user name given to the document and the name by
14 which the document is known to the user.

15 Further, document properties 276a-276n can include an author identifier, such as a
16 document distributor attribute and/or a free storage attribute. These attributes identify
17 whether or not a general user or document distributor creates the document.

18 The document properties 276a-276 can further include a default version number
19 defining the current version of the document that is opened by default. Because multiple
20 versions of each document can be stored, such as described in United States Patent No.
21 6,181,080, it can be desirable to store a default version number so that application
22 programs such as control application 232 (Figure 3) know which version to retrieve by
23 default when opening or editing a document.

24

1 In addition, document properties 276a-276n can include a global ID of the
2 document. The global ID is a unique identifier given to each document to aid in matching
3 documents in various copies of container files 240a-240n. A global ID is useful, for
4 example, when trying to synchronize and resolve conflicts between two copies of a
5 container file.

6 Further, the identity of the last user to save the document can be tracked and
7 included as another attribute of document properties 276a-276n. Such an attribute can
8 have a value or identify the last user to save the document. Similarly, the last saved
9 date/time for the document can be stored within document properties 276a-276n.

10 In addition to the above, document properties 276a-276n can include a documents
11 collision group. The document collision group is a property helpful in handling conflicts.
12 For example, if different individuals change two copies of a document simultaneously, a
13 conflict between the versions can exist. In this case, it can be possible to save both
14 versions of the document and allow a user to sort it out. The document can then be
15 identified as a member of a collision group until the conflict is resolved.

16 Another illustrative property of document properties 276a-276n is a document's
17 tamper seal. The tamper seal is information that allows unauthorized access or tampering
18 of the document to be recognized. For example, such a tamper seal can include a check
19 sum or digital fingerprint of the information calculated according to some method which
20 allows tampering of the document to be identified. In one embodiment the MD5 algorithm
21 is used to generate a signature. If multiple versions are stored, a tamper seal for each
22 version can be stored.

1 In addition to document properties 276a-276n, Figure 4 illustrates that each
2 document storage 270a-270n includes document contents 278a-278n. The document
3 contents 278a-278n is the content to be displayed or otherwise accessed by user 212
4 (Figure 3). For instance, when document storage 270a-270n is a word processing
5 document, document contents 278a-278n include the words and optionally graphics that
6 are displayed to the user when a document is accessed by way of a word processing
7 application. Similarly, when document storage 270a-270n is a spreadsheet, document
8 contents 278a-278n include the data, formulae, and the like to be inserted into each cell of
9 the spreadsheet. Various types of document contents 278a-278n are applicable for use
10 with the present invention and known to one skilled in the art in light of the teaching
11 contained herein.

12 As illustrated in Figure 4, each document storage 270a-270n can include an
13 optional tracking log 280a-280n. As explained above, tracking log 266 stored at root
14 storage 260 contains information regarding events that have occurred for all documents
15 within container file 240a. However, storing a single large tracking log at the root storage
16 level is not the only way to record and track events that occur to container file 240a or to
17 documents 270a-270n within container file 240a. For example, each entry in a tracking
18 log can include a wide variety of information.

19 The information tracked by tracking logs 266 and 280a-280n can include, for
20 example, the identity of the individual taking the action, the action that occurred, the object
21 of the action, the time that the action occurred, and the like. Organizing these entries can
22 occur in a wide variety of ways. For instance, it can be desirable to organize the event
23 entries by actions. As another example, it can be desirable to organize the event entries by
24

identity of the user taking the action. As yet another example, the event entries can be organized by the target of the action. In such a situation, it can be desirable to maintain individual tracking logs for each of the various documents rather than one global tracking log for all entries. It can also be desirable to maintain some combination of the two.

As illustrated in Figure 4, tracking logs 280a-280n are shown in dashed lines to indicate that such a log can be optional, depending upon the particular implementation selected. In addition, it may not be necessary to track global events and thus may not be necessary to maintain tracking log 266, depending on the particular implementation. In general, however, it is believed that it is important to track events that occur to the container file itself. In such an embodiment, it is desirable to store such events in a global tracking log such as tracking log 266.

Referring now to Figure 5, a flow diagram representing one illustrative manner by which documents are stored within document storage repository 206 is depicted. One skilled in the art can identify various other manners for storing documents within document storage repository 206. For instance, United States Patent No. 6,181,080 discloses a manner to deliver documents to and from a document storage repository by various delivery transports, such as e-mail, or the like.

As shown in Figure 5, a user begins by creating a document, as represented by block 290. The user can create the document employing one or more application programs, such as a word processing application, a spreadsheet application, a graphics application, and the like. Typically, user 212 (Figure 3) can create a document and store the same upon a computer, such as computer 20 (Figure 1).

Following creation of the document, the user can attempt to access document storage repository 206, as represented by block 292. As discussed above, this can include initiating a web browser on a workstation and connecting with a web site by inputting a uniform resource locator (URL) identifier. Alternatively, and more typically used by a document distributor, a server can directly communicate with document storage repository 206 (Figure 3) by way of an API.

7 As part of the process to access document storage repository 206, the user is
8 authenticated by document storage repository 206, as represented by block 294. In one
9 configuration, the authentication process is required when the user accesses document
10 storage repository 206 via a workstation or when a server attempts to automatically
11 communicate with document storage repository 206. In the former case, such
12 authentication can entail the user inputting a user name or ID and password into a
13 graphical user interface associated with control application 232. In a similar manner, the
14 server can upload an appropriate user name or ID and password to document storage
15 repository 206 for authentication. One skilled in the art can identify various other manners
16 for securely providing authentication data to document storage repository 206.

17 Once document storage repository 206 has received a user name and password, it
18 verifies such information against the user information stored within directory module 234.
19 Specifically, control module 230 compares the user name or ID and password against
20 stored values for the appropriate user included in the lists of users in directory module 234.
21 If the information received from the work station and/or server matches the information
22 stored within directory module 234, the user is authenticated and allowed to access
23 document storage repository 206. Alternatively, in the event the user is not authenticated,

as represented by decision block 296 being in the negative, the user is denied access to document storage repository 206 and is unable to store the document or any other information therein or otherwise access any previously stored document or container file.

Upon authenticating the user, control module 230 identifies whether or not the user is a document distributor, as represented by decision block 300. In the event that the user is not a document distributor, control module 230 identifies the container files stored within storage module 238 that are accessible by the user. Consequently, control module 230 can present a graphical user interface that allows the user to drag and drop the particular document into a graphical representation of one or more of the container files associated with the user, as represented by block 308. Various options are provided depending on whether the user is a document distributor or not. In the event the user is not a document distributor, the user can store the document within the user's account or container file. Following such storage, the work lists stored in storage module 238 are updated to identify the document accessible by the specific user, as represented by block 310. Further, container file attributes can be modified, such as updating the date when the container file was last modified, the data stored therein, and the like. Following the storage of the document, the user can continue to access document storage repository 206 or alternatively disconnect or "log out" of document storage repository 206.

In the event that a user is a document distributor, as represented by decision block 300 being answered in the affirmative, control module 230 identifies the particular user that the document is to be delivered to and consequently stores the document in such a user's account, as represented by block 302. Optionally, the document distributor can

automatically insert the document into a plurality of user accounts by defining an access list associated with the access properties of the document.

Substantially simultaneously with delivery of the document to the various users, or alternatively before or after delivery of the document and associated container file to one or more users, worklist module 242 is updated with information regarding to whom the document was delivered, the size of the document that was delivered, and/or the number of documents delivered to the various users, as represented by block 304. After updating worklist 242, the temporary transaction log is updated with information regarding the activities of the user, whether or not the user is a document distributor or a general user.

Although it is discussed above that a document distributor can automatically insert a document into another user's account, it is possible for other users to deliver documents to other users. While such an operation does not automatically insert the document into another user's account it does provide the functionality for sharing and delivering documents between various other users. For instance, in one configuration, a user can take the document they have created or have access to and insert the same into a container file deliverable to another user, *i.e.*, a deliverable container file. This container file is then delivered to the recipient user that may access this container file and associated document(s). The delivering user, by delivering the container file and document(s) to the recipient relinquishes control of the container file and associated document(s). It is, however, possible for the delivering user to maintain control of the container file and document(s) by sharing the document(s) with other users via a shared container file. In this manner, the user can track the actions of other users as they use or manipulate the document(s) contained within the shared container file.

1 The above-described method is also schematically illustrated in Figure 6, which
2 represents the internal modules, components, attributes, and/or properties described above
3 with respect to document storage repository 206 and the illustrative communication and
4 linking therebetween. Although single modules, components, and attributes are depicted
5 in Figure 6, one skilled in the art can appreciate that multiple modules, components, and
6 attributes can be included and are considered part of this invention. Similarly, the multiple
7 modules, components, and attributes depicted in Figure 6, can take the form of single
8 modules, components, and attributes.

9 As discussed above, control application 232, whether the GUI or the API can
10 receive requests for access to document storage repository 206. Following receipt of the
11 access request, control application 232 communicates with directory module 234 and
12 accesses information about the user accessing document storage repository 206.
13 Specifically, control application 232 accesses a user list, designated by reference number
14 320. Upon accessing the user list, specific information about the user requesting access to
15 document storage repository 206 is retrieved. For instance, let us assume that User 2 is
16 requesting access to document storage repository 206. Consequently, control application
17 232 retrieves User 2 properties, designated by reference number 322, and compares the
18 received user name or ID and password against the stored user name or ID and password.
19 If they match, access is allowed, otherwise access to document storage repository 206 is
20 denied. It should be understood that the user name/password authentication discussed
21 herein is just one illustrative example of the type of authentication information that could
22 be used in the present invention.

1 Following authentication, control application 232 and/or control module 230 can
2 receive document(s), facilitate the exchange or delivery of documents and container files
3 between one or more users, share documents and container files with multiple users, or the
4 like. For instance, let us assume that the user wishes to insert a document into their user
5 account, *i.e.*, a container file designated for the users personal use, such as Container
6 File 3. As shown in Figure 6, as a user “logs in” to document storage repository 206,
7 control application 232 and/or control module 230 retrieves the appropriate container file
8 326 from a container file list, designated by reference numeral 324. Such a container file
9 could be, for example, Container File 3. Subsequently, the properties and other
10 information related to Container File 3, as depicted by reference number 326, are retrieved
11 so that Container File 3 or a graphical or textual representation of Container File 3 can be
12 presented to the user through control application 232.

13 In the event that a user wishes to store a document, such as Document 1, the user
14 can, in one configuration, drag an icon representation of Document 1 into the
15 representation, whether textual and/or graphical, of the user’s personal container file.
16 Consequently, a document 328a-328n is created within Container File 3 and the document
17 content and document properties are created, whether automatically by control application
18 232 and/or control module 230, manually by the user inputting information into the GUI
19 presented to the user by control module 230, or a combination thereof.

20 These document properties can include a designation that the author of the
21 document is a document distributor. This can be achieved through a document distributor
22 attribute having a pointer to the particular user entry within directory module 234, or
23 alternatively, through a document distributor attribute having a “true” or “false” value.

1 Further, the document can include a free storage attribute that identifies that the document
2 was created by a document distributor, while acting as a tag or notifying attribute that the
3 amount of space used by the document is be excluded from the calculated storage total for
4 the account within which the document is currently stored.

5 In addition, properties of Container File 3 can be updated to reference the inclusion
6 of a new document, say document 328a-328n. Therefore, container properties 330 can be
7 updated to reference a change to the content of the container file, identify who updated the
8 container file, and the like. In a similar manner to document properties, container
9 properties can include a designation that the author of the document is a document
10 distributor by optionally including a document distributor attribute and/or a free storage
11 attribute.

12 In some situations, the user, whether a document distributor or another user may
13 wish to share the document stored within Container File 3 with other users. For instance,
14 during a project an individual may wish to share multiple documents with a variety of
15 individuals. To achieve this, the user can optionally modify the access properties 332 for
16 Container File 3. The access properties 332 may define a list of one or more users that can
17 access Container File 3. Therefore, any user listed in access properties 332 can retrieve or
18 access the documents contained within Container File 3. Limited access can be provided
19 based upon some restriction that might be referenced with respect to the access properties.
20 For example, some users may be allowed to read or write to documents, while other users
21 may only be allowed to read documents.

22 In an alternate configuration of the present invention, the user can share documents
23 via a different manner. In such a case, the user can create another container file within
24

1 which the user inserts or stores those documents to be shared between the various users.

2 As before, the user defines the access properties to control the access to those documents.

3 A similar process can be followed to access documents stored within a container file, such

4 as Container File 3.

5 Further, as a user desires to view a container file and/or a document stored therein,
6 control application 230 can retrieve logo and/or trademark information, if applicable, from
7 the user that authored the container file and/or document. For example, say a document
8 distributor authored a container file and/or documents included therein that is stored by a
9 user. Upon accessing the container file and/or the documents, control application 232
10 identifies the author from the container properties and/or the document properties and
11 retrieves the logo and/or trademark from the user account information associated with the
12 document distributor. Control application 232 can subsequently display the logo and/or
13 trademark to the user as they access the documents and/or container files authored by the
14 document distributor. It can be appreciated by one skilled in the art that there are a number
15 of different ways by which document distributor information can be presented to a user
16 that accesses a container file and/or document authored by the document distributor.

17 Referring now to Figure 7, depicted is a flow diagram illustrating a method by
18 which resource allocation analysis is performed according to one aspect of the present
19 invention. This particular method is illustrative of one particular manner by which storage
20 allocation information can be collected by document storage repository 206. The
21 calculation of resources used can be performed at various times by control module 230
22 whether alone or in combination with resource allocation module 250 (Figure 3). For

instance, such resource allocation analysis may be performed continuously, periodically, in accordance with user defined instructions, a combination thereof, or the like.

Initially, control module 230 retrieves a worklist from worklist module 242 (Figure 3) containing information about each user and those container files and/or documents accessible by the user, as represented by block 350. In one configuration, control module 230 and/or resource allocation module 250 retrieves the information from the worklist and the access information specific to each user. For instance, upon retrieving the worklist, control module 230 and/or resource allocation module 250 can identify each container file and/or document within a user's account that are accessible by the user. Further, control module 230 and/or resource allocation module 250 can access each container file and/or document, as represented by block 352.

As control module 230 and/or resource allocation module 250 accesses each container file and/or document, the value of the document distributor attribute or free storage attribute for the container file and/or document is retrieved, as represented by block 354. These attributes can be pointers, true or false values, or the like. It is understood by one skilled in the art, in light of the disclosure contained herein, that the designation of a container file, or for that matter a document, as authored by a document distributor can be performed in a variety of manners.

In the event that the document distributor attribute or the free storage attribute has a value, such as true, as represented by decision block 356, the storage total and/or the number of documents stored within the container file is excluded from the resource allocation total for the user that is currently storing the container file and/or documents, as represented by block 360. Alternatively, if the document distributor attribute or the free

1 storage attribute is false or there is no value, the storage total and/or the number of
2 documents stored within the container file is included in the resource allocation total for
3 the user currently storing the container file and/or documents, as represented by block 358.

4 In another configuration, the document distributor attribute or the free storage
5 attribute is a pointer to a particular user entry within directory module 234 (Figure 3).
6 Consequently, if the document distributor attribute or the free storage attribute includes a
7 pointer, control module 230 and/or resource allocation module 250 tracks the pointer to the
8 designated user and checks a corresponding document distributor attribute associated with
9 the user's properties. In the event that such a document distributor attribute designates the
10 user as a document distributor, the storage total and/or the number of documents stored
11 within the container file and/or document that "pointed" to the document distributor is
12 excluded from the resource allocation total for the user that is currently storing the
13 container file and/or documents, as represented by block 360.

14 In contrast, in the event that the document distributor attribute for the "pointed to"
15 user does not designate the user as a document distributor, the storage total and/or the
16 number of documents stored within the container file are included in the resource
17 allocation total for the user currently storing the container file and/or documents.

18 Stated in another way, if an entry in worklist module 242 (Figure 3) that is
19 currently being analyzed is for a first user, and if the first user is not a document
20 distributor, but the documents and/or containers stored within the first user's account are
21 authored by a document distributor, the total size and/or number of documents and/or
22 container files stored within the first user's account that are authored by a document
23 distributor are excluded from the first user's storage total. Consequently, this first user

1 would not be allocated a usage total including the documents and container files received
2 and authored by a document distributor. In contrast, the usage total for the corresponding
3 document distributor would include the size and/or number of such documents and/or
4 container files sent to the first user.

5 The above process continues until all the container files and/or documents within
6 the worklists of worklist module 242 (Figure 3) are checked, as depicted by decision block
7 362. Once all container files and/or documents within worklist module 242 (Figure 3) are
8 checked, the storage or usage total for each user is calculated, as represented by block 364.

9 Referring now to Figure 8, an alternative illustrative method by which storage
0 resources are allocated and calculated is depicted. More specifically, the flow diagram of
1 Figure 8 illustrates a method by which bills can be calculated for storage usage by each
2 document distributor where control module 230 and/or resource allocation module 250 has
3 maintained an ongoing list of the number of documents and/or container files that each
4 document distributor sends to other users. In this particular method, control module 230
5 (Figure 3) optionally in combination with resource allocation module 250 selects a specific
6 document distributor for which resource allocation is to be calculated. Following selection
7 of the document distributor, as represented by block 370, the number of documents and/or
8 container files delivered to other users is calculated, as represented by block 372.
9 Specifically, resource allocation module 250 retrieves information stored within temporary
0 file 252 and updates the data stored within resource allocation module 250. Resource
1 allocation module 250 then identifies, by checking a tracking log stored within resource
2 allocation module 250 the number of documents and/or container files delivered to other
3 users.

1 Optionally, the calculation can include a determination of the total amount of
2 storage allocated to a particular document distributor instead of identifying the number of
3 documents and/or container files delivered to various users. Upon calculating the total
4 number of documents, container files and/or amount of storage used, resource allocation
5 module 250 can calculate an appropriate bill to be charged to the document distributor, as
6 represented by block 374.

7 Referring now to Figure 9, depicted is another illustrative method by which used
8 storage capacity of document storage repository 206 can be allocated to a document
9 distributor instead of the recipient that receives documents and/or container files from the
10 document distributor. Specifically, a worklist for each user is opened, as represented by
11 block 380. As discussed previously, each worklist of worklist module 242 (Figure 3)
12 represents the container files and documents that are accessible by a specific user, whether
13 or not such documents are authored by the user, received by the user, shared by the user
14 and other users, and the like. Once the worklist for a specific user is identified, each
15 container file and/or document identified within the worklist is opened. Specifically, a
16 container file and/or document is opened, as represented by block 382, and subsequently
17 the container properties and optionally document properties for each document stored
18 therein are checked to determine whether or not the container file and/or documents are
19 authored by a document distributor, as depicted by block 384. This process is continued
20 until all container files and/or documents specific to a user are identified and analyzed, as
21 represented by decision block 386.

22 The process of opening the container files and/or documents and consequently
23 determining whether or not the container file and/or document are authored by a document